UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,396	07/07/2003	Lin Yan	10016253-1	5944

22879 7590 02/08/2008

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
ZHE, MENG YAO	
ART UNIT	**PAPER NUMBER**
2195	

NOTIFICATION DATE	DELIVERY MODE
02/08/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

MN

Office Action Summary	Application No.	Applicant(s)	
	10/614,396	YAN ET AL.	
	Examiner	Art Unit	
	MengYao Zhe	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26 and 28-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26, 28-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/ are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-26, 28-34 are presented for examination.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 1-26, 28-34 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. The following claim languages are unclear and indefinite:

i) As per claims 2, 3, 12, 13, it is unclear what is meant by "a DLKM type identifier" <i.e. abbreviations are not allowed>.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent

granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 9, 10, 30, 31, and 34 are rejected under 35 U.S.C. 102(e) as being anticipated by Berg et al., Patent No. 6449660 (Hereafter Berg).

4. As per claim 1, Berg teaches a method for maintaining a module type definition table *(Column 16, lines 55- 60; Column 14, lines 8-21; Column 14, lines 43-60)* by a statically configured portion of an operating system kernel, comprising:

Identifying a module type (class information is the module type) *(Column 21, lines 15-33)*

Searching an external module type definition table for the module type *(Column 16, lines 55-60; Column 14, lines 8- 21; Column 14, lines 43-60)*

Determining the module type is not defined in the module type definition table *(Column 21, lines 32-34)*

Dynamically creating a module type definition *(Column 21, lines 28- 41)*;

Updating an external module type definition table to include the module type definition at the direction of the static operating system kernel *(Column 21, lines 28-40)*.

5. As per claim 9, Berg teaches:

Wherein creating a module type definition includes receiving at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with a support module *(Column 14, lines 49-51; Column 17, lines 27-40: the child device corresponds to the support module.)*

6. As per claim 10, Berg teaches:

creating a module type definition includes receiving at least one symbol name, each symbol name being respectively associated with a support module *(Column 14, lines 43-50; Column 17, lines 25- 40)*

7. As per claims 30 and 31, it has all the logics that are also contained in claim 1. Since claim 1 is rejected, claims 30 and 31 are rejected as well.

8. As per claim 34, Berg teaches logic on a computer readable medium to identify at least one support module associated with the module type. (Column 14, lines 44-59)

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 4-8, 11 and 14-26, 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of: The Windows NT Device Driver Book by Art Baker, 1997 (hereafter Baker).

11. As per claim 11, Berg teaches a system for maintaining a module type definition table *(Column 14, lines 13-16)*, comprising:

module type detection logic on a computer readable medium for detecting that a module is of an undefined module type; *(Column 15, lines 5-24; Column 21, lines 32-34: the names used to identify the device including Rtok, RscName correspond to module type.)*

module type identification logic on a computer readable medium for assigning a new module type associated with the module; *(Column 21, lines 32-34)*

support module identification logic on a computer readable medium for identifying at least one support module associated with the module; *(Column 14, lines 45- 53; Column 17, lines 24-40: the child device corresponds to the support module.)*

module type definition logic on a computer readable medium for

externally storing data defining the module type. *(Fig 8, unit 813 is external to the operating system.)*

12. Berg does not teach support module loading logic on a computer readable medium for loading the at least one identified support module. However, Baker teaches loading one identified support module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver may specify all the modules it depends on, which corresponds to the support modules, for them to be loaded.)

13. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg, with: support module loading logic on a computer readable medium for loading the at least one identified support module, because it allows the user to created load-order dependency.

14. The Examiner notes that the system as claimed in 11 has the capabilities to do all of the method steps found in claim 1.

15. As per claim 14, Baker teaches a support module operative to conduct pre-registration support. (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the first driver that needs to be loaded before the second driver corresponds to the pre-

registration support module. The Examiner considers the pre-registration support module to be the equivalent of pre-loading module)

16. As per claim 15, Baker teaches a module to conduct a registration function *(Pg 414, third bullet: in the case that the second module is dependent on the first module, the DependOnGroup specifies names of modules that the second module depends on. This corresponds to registration.)*;

17. As per claim 16, Baker teaches a post-registration module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver that needs to be load after the second driver is considered to be the post-registration module.)

18. As per claim 17, Baker teaches a support module operative to conduct pre-loading support. (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the first driver that needs to be loaded before the second driver corresponds to the pre-loading support module.)

19. As per claim 18, Baker teaches a post-loading module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to

modules, that need to be loaded, the third driver that needs to be load after the second driver is considered to be the post-loading module. The Examiner considers the post-registration support module to be the equivalent of post-loading support.)

20. As per claim 19, Berg teaches:

> Wherein creating a module type definition includes receiving at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with a support module *(Column 14, lines 49-51; Column 17, lines 27-40: the child device corresponds to the support module.)*

21. As per claim 20, Berg teaches:

creating a module type definition includes receiving at least one symbol name, each symbol name being respectively associated with a support module *(Column 14, lines 43-50; Column 17, lines 25- 40)*

22. As per claim 21, it contains all the instructions necessary to perform the method steps capable by the system of claim 11. Since claim 11 is rejected, claim 21 is rejected as well.

23. As per claim 26, Berg teaches a static operating kernel on a computer readable medium (Fig 8) comprising:

logic on a computer readable medium to receive a request to load a module (Column 14, lines 55-60)

logic on a computer readable medium to identify a module type of the module (Column 21, lines 15-26)

logic on a computer readable medium to reference an external module type definition table (Column 14, lines 10-20)

logic on a computer readable medium to identify at least one support module associated with the module type in the external module type definition table, (Column 14, lines 43-50: children corresponds to support modules)

logic on a computer readable medium for Identifying at least one module type net previously defined in the external module type definition table (Column 21, lines 30-36)

24. Berg does not teach logic on a computer readable medium to load the module based upon the module type and the at least one support module associated with the module type. However, Baker teaches logic on a computer readable medium to load the module based upon the module type and the at least one support module associated with the module type (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third

driver may specify all the modules it depends on, which corresponds to the support modules, for them to be loaded.)

25. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg, with: logic on a computer readable medium to load the module based upon the module type and the at least one support module associated with the module type, because it allows the user to created load-order dependency.

26. As per claim 28, Berg teaches the logic to dynamically define the at least one external module type includes receiving an operator identified module type. (Column 14, lines 55-60: the administrators corresponds to the operators. They are allowed to add new devices.)

27. As per claim 29, Berg teaches the logic to dynamically define the at least one external module type includes receiving at least one identified support modules from an operator. (Column 14, lines 43 to 60: In the case that the administrator adds a device that is classified as a children, it would corresponds to a support module.)

28. As per claim 4, it contains all the method steps that may be performed by the system of claims 11 and 14. Since claims 11 and 14 are rejected, claim 4 is rejected as well with the same motivation.

29. As per claim 7, it contains all the method steps that may be performed by the system of claims 11 and 17. Since claims 11 and 17 are rejected, claim 7 is rejected as well with the same motivation.

30. As per claim 22, it contains all the instructions that may be carried out by the system of claim 17. Since claim 17 is rejected, claim 22 is rejected as well.

31. As per claim 24, it contains all the instructions that may be carried out by the system of claim 14. Since claim 14 is rejected, claim 24 is rejected as well.

32. As per claim 5, it contains all the method steps that may be performed by the system of claims 11 and 15. Since claims 11 and 15 are rejected, claim 5 is rejected as well with the same motivation.

33. As per claim 6, it contains all the method steps that may be performed by the system of claims 11 and 16. Since claims 11 and 16 are rejected, claim 6 is rejected as well with the same motivation.

34. As per claim 8, it contains all the method steps that may be performed by the system of claims 11 and 18. Since claims 11 and 18 are rejected, claim 8 is rejected as well.

35. As per claim 23, it contains all the instructions that may be carried out by the system of claim 18. Since claim 18 is rejected, claim 23 is rejected as well.

36. As per claim 25, it contains all the instructions that may be carried out by the system of claim 16. Since claim 16 is rejected, claim 25 is rejected as well.

37. Claims 12, 13, 32 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of: The Windows NT Device Driver Book by Art Baker, 1997 (hereafter Baker) further in view of Managing and Developing Dynamically Loadable Kernel Modules (hereafter HP), Copyright 2001, Hewlett-Packard Company.

38. As per claim 12, Berg in view of Baker teaches the invention substantially as claimed including all of claim 11.

39. Berg in view of Baker does not teach receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier. However, HP teaches a

demand load DLKM for the purpose of providing a user level request for a specific module to be loaded (Chapter 12, page 501: *The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is demand load, the Examiner considers this to be the equivalent of operator generated type identifier.)*

40. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg in view of Baker with receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier, as taught by HP (Chapter 12, page 501)

41. As per claim 13, Berg does not teach receiving a computer generated dynamically loadable kernel module type identifier.

42. However, HP teaches an autoload DLKM *(Chapter 12, page 502)*; *(The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is autoload, the Examiner considers this to be the equivalent of computer generated type identifier.)* Refer to claim 12 for the motivation to combine.

43. As per claim 32, since claim 12 is rejected, and with the same motivation to combine the teachings of Berg in view of Baker and HP, claim 32 is rejected as well. (the operator generated module corresponds to operator generated DLKM)

44. As per claim 33, since claim 13 is rejected, and with the same motivation to combine the teachings of Berg in view of Baker and HP, claim 33 is rejected as well. (the operator generated module corresponds to computer generated DLKM)

45. Claims 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of Managing and Developing Dynamically Loadable Kernel Modules (hereafter HP), Copyright 2001, Hewlett-Packard Company.

46. As per claim 2, Berg teaches the invention substantially as claimed including all of claim 1.

47. Berg does not teach receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier. However, HP teaches a demand load DLKM for the purpose of providing a user level request for a specific module to be loaded (Chapter 12, page 501: *The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a*

module type definition must be coupled to the DLKM. Since the status is demand load, the Examiner considers this to be the equivalent of operator generated type identifier.)

48. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg in view of HP with receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier, as taught by HP (Chapter 12, page 501)

49. As per claim 3, Berg does not teach receiving a computer generated dynamically loadable kernel module type identifier.

50. However, HP teaches an autoload DLKM *(Chapter 12, page 502)*; *(The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is autoload, the Examiner considers this to be the equivalent of computer generated type identifier.)* Refer to claim 2 for the motivation to combine.

Response to Arguments

51. Applicant's arguments filed on 11/15/2007 have been fully considered but are not persuasive.

In the remark, the applicant argued that:

i) Berg does not teach dynamically creating a module type definition that includes at least one support module identifier.

ii) The child device, as taught by Berg, is not a support module.

The Examiner respectfully disagree with the applicant. As to point:

i) Berg teaches an instantiating step (Column 21, lines 20-41) for multiple DASD devices commonly connected through a DASD interface (DASD device corresponds to a module) after failure to map DASD device Uids to existing Uids (DASD device Uid corresponds to a type of module type definition). The instantiating step includes initializing configuration data which includes RscName for the device, which the Examiner considers to be a way of creating module type definition. The applicant merely pointed out that the step of initializing configuration data (which the Examiner equates to module type definition) for the new device (which the Examiner equates to a module) is not the equivalent creating module type definition without explaining why or how those two steps are not the same.

In addition, applicant has not specifically disclosed to what degree or how this operation is performed dynamically. Boot time is the time when a computer system starts up and initializing its programs and resources. This theoretically is dynamic since the operations are performed when the system is running, similar to how Berg functions. Berg further starts child devices and connects them by notification to the parent/controller device. Therefore, a support module identifier is created when the device is instantiated. The claim language of dynamically creating a type definition is not limited to creation of non-static identifiers, but also includes when static identifiers are created in memory, e.g. at runtime.

Furthermore, Berg also teaches that the DASD interface contains VPD that includes all important information about other devices that are connected to or related to the DASD device. The VPD information is relayed to all relevant devices. The Examiner equates all related devices of the newly instantiated DASD device to be the support modules (Column 21, lines 47-65). The applicant argued that that in the specification, a support module may, for example, "be used to provide module type-specific support for the identified module...", however, this is merely an example of what a support module may be, and not a definition of what it is (see MPEP 2131.05). Therefore, the definition of a support module is open to interpretation.

ii) The Examiner has the freedom to interpret the child device that is related to a newly instantiated device to be the support module of that newly instantiated device. The applicant argued that that in the specification, a support module may,

for example, "be used to provide module type-specific support for the identified module...", however, this is merely an example of what a support module may be, and not a definition of what it is (see MPEP 2131.05). Therefore, the definition of a support module is open to interpretation, which could be a child device invoked by the parent to handle an operation and provides support to the parent by performing that operation, and this corresponds to a support module.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MengYao Zhe whose telephone number is 571-272-

6946. The examiner can normally be reached on Monday Through Friday, 10:00 - 8:00 EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached at 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER